CHAPTER

8

PERFORMANCE MONITORING

After completing this chapter, you will be able to:

- Describe the tools you can use to monitor system performance
- Describe the function and purpose of System Monitor
- Describe and use counters in System Monitor to analyze system performance
- Describe the purpose of the Event Viewer
- Configure alerts in System Monitor
- Define the logs that exist on a Windows 2000 domain controller
- Describe the tools available to monitor Active Directory performance

Domain controllers (DCs) have a significant role to play on your Windows 2000 network. What's more, problems that occur at DCs can show up in the strangest places. Perhaps you are getting calls from your users, complaining that the network is slow; or maybe users are having trouble saving documents to a particular share or logging on. DCs have such an important role to play that problems with a single one can manifest itself in hundreds of different ways.

In this chapter, you will learn about monitoring your domain controllers. You will see that making sure they are operating correctly is a common task that must be performed on a regular basis. We will discuss the tools and techniques you can use to maintain an efficient network.

Introduction to Performance Monitoring

If you asked 10 professionals what characteristics a fast computer should have, you would probably get 10 different answers—what is important to one person has no meaning to another. When it comes to Information Technology (IT) professionals, the truth is that we are not particularly good at defining what a good system really is.

There are several reasons for this. First, a system configured to operate in one environment, to answer one specific need, may not meet the requirements of another environment. This fact explains why a Web server may work well until you install Microsoft Exchange or some other messaging system on it—what works for one function just won't work for another. Second, applications have their own stress points and things you can do to optimize them. Without an awareness of these features, you cannot possibly make intelligent design decisions.

For some administrators, the goal is always to have the least amount of work going on at a particular server. The Nirvana is assumed to be processors that are never stressed beyond 10% utilization and memory capacity that is never stretched beyond 50% usage. We would like to dispel that myth. Think of it in terms of a car. What is the point in buying a Maserati sports car, if you're going to coast in the slow lane at 20 miles per hour? You are essentially doing the same thing when you buy a multiple-processor-enabled server, with RAID controllers and gigabytes of RAM, only to use it for file sharing.

That is not to say that a neat list exists of what is acceptable and not acceptable. If a server is operating at 40 percent of processor capacity all day and night, however, is that really a bad thing? As long as it is fulfilling its role, the answer may very well be "no." You should strive to get a handle on your servers and what they are doing. But the goal is not to eliminate load completely—it is to control the load, and to identify when something out of the ordinary is taking place.

Many different things can affect the performance of a computer. Most people concentrate their efforts in the areas of processors and memory. We have been sold this approach for years now in magazine articles and white papers: Add processors (or put in a new processor with better performance), add some more Random Access Memory (RAM), and things should be fine. Many of us have found, after filling our Windows 95 and Windows 98 boxes with 256 MB of RAM, that at a certain point, more does not equal better.

Of course, we are not talking about Windows 95 or Windows 98 here; we are talking about Windows 2000. Like all new operating system releases, this latest release requires newer hardware to fully reach its potential. Although minimum requirements do not look dissimilar from previous versions of Microsoft Windows NT, don't be fooled into thinking that a fast Pentium III with at least 256MB of RAM won't make things run a lot faster—because they will!

So, in a world where finances are finite, and where your managers are asking you to cut costs, what can the system administrator do to make sure you are getting the best bang

for the buck? To begin with, you can get a better understanding of how systems perform, and why they react the way they do to various events.

Performance monitoring is an important skill, because by using it you can get a **signature** for the servers in your environment. A signature is also known as a **baseline**, which can be further defined as a level of performance on a normal working day. The baseline defines how the server will operate when things are running normally. Once you have this information, you can then run performance statistics periodically and compare the results. Are things running efficiently? Are there deviations from the baseline? Are problems developing, and in what areas are you having problems? All these questions and more can be answered if you know which performance statistics are available to you, what they mean, and how to use them. Introducing you to these statistics is the purpose of this chapter.

Furthermore, we are going to look at several of the tools at your disposal. One of the most important tools is System Monitor. System Monitor allows you to access counters that can report on system performance.

We could list every counter available to you and define what they do, but doing so would simply fill a lot of pages with boring lists. Instead, we will define some critical counters for a DC operating in a Windows 2000 environment. We'll define some important terms in a moment so you don't get lost.

Performance monitoring (which, incidentally, goes beyond simply using the System Monitor tool in Windows 2000) hardly ever gets its due attention. All those counters can be perplexing; once you have the data, moreover, you have to make sense of it. This chapter will walk you through what performance monitoring really is, how it works, how it differs from what we have seen in previous versions of Windows NT, and how you can use these tools to predict problems before they occur, thus avoiding costly outages. This may not be the most exciting chapter in this book, but the rewards you can get from understanding this topic could pay you back many times over.

Windows 2000 System Monitor

System Monitor, also commonly known as simply perfmon, has been included with Microsoft Windows NT for a number of years. The version that ships with Windows 2000 has been improved in some significant ways, so even if you are familiar with the tool, you should take time to read through this section. The default System Monitor display is shown in Figure 8-1.

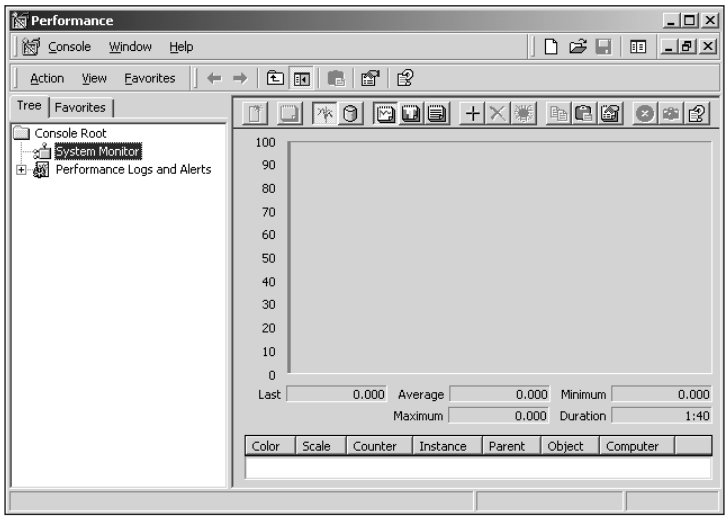

Figure 8-1 System Monitor

Before we dig into the details, let's define what it is that you want to monitor. A server is a complex mix of components, and simply monitoring the hardware characteristics won't be enough. In fact, you need to be concerned with all aspects of a server's performance including hardware, applications, the operating system, and anything else that has been enabled for monitoring.

The true measure of a system's performance is usually a combination of all of these things. You can have state-of-the-art hardware with horrible software, and you will be unhappy with performance. Or, you can buy and install the latest and greatest operating system and put it on an older piece of hardware, and you just won't get that cutting-edge feel you hoped for.

When you examine a server to determine its performance, you'll look in four key areas:

- Memory
- Processor
- Disk
- Network

These areas obviously are key to a server—if any one of them is significantly lacking, then you can expect to see degraded performance. Measuring them will become an important part of your skill set. Let's briefly describe each of these areas.

Memory

Memory is the RAM inside the server (or any other machine being monitored). Memory usage is actually quite complex. When you look at memory usage on a Windows 2000 server, you generally simply glance at the Physical Memory counter on the Performance tab of Task Manager. However, this value provides a very high-level view of memory and how it is being used. It can let you know immediately whether a server needs more RAM, but it does nothing to tell you about memory usage over time, or what process is using the memory.

For the purposes of setting a baseline or discovering the performance characteristics of a given server during normal use, you need to become more familiar with processes that are running on a server and how they are used. We'll take a closer look at some of the important memory counters later in this chapter. They will give you an idea of some common areas of concern, and how you can monitor potentially problematic situations. Figure 8-2 shows some of the counters available for the Memory object, as shown in System Monitor.

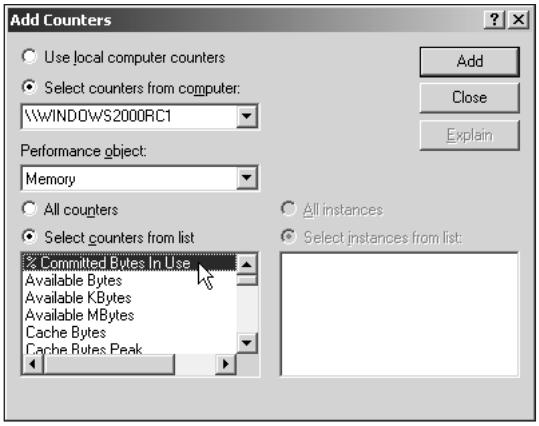

Figure 8-2 The counters of the Memory object

When a server runs low on memory, it begins to use hard disk space as though it were physical memory. This process is called **paging**. If you ever had a bad day when everything on your system was slow and saving documents to your hard disk took forever, you were probably paging. Paging is efficient, in that your system does not crash simply because you have a lack of memory; but it is also very slow. Think of paging as a cry for help. We'll show you how to identify this common problem.

Processor

As often as you will hear technical support cry, "Have you tried rebooting it?" you will hear your IT staff say "I need a faster processor!" There is no denying the fun in having the latest and greatest technology from Intel or AMD—but the fact is, we underutilize most processors.

An underpowered processor that is straining to keep up with all the work you throw at it is going to hold up every other component on a system, however. Of course, you can upgrade processors, or add more than one. We will take a look at processors later.

Disk

The disk subsystem of your server is probably the most overlooked component you have. However, if you corner a database person (maybe a Microsoft SQL Server 2000 or Oracle DBA expert) and ask him where most performance issues can be solved, he'll start to talk at length about disk subsystems.

Data is read into a computer's memory, worked on by code and the processor, and then generally written somewhere on the local hard disk before being either transported somewhere else or saved permanently. Two of those three processes—the reading into memory and manipulation by the processor—are extremely fast. Writing out to disk, however, is a mechanical process, and you can quickly run into a lot of trouble.

You must know how quickly data is being written to and read from the disk. Are things waiting to be copied to the disk? How quickly does the disk respond? These aspects will have an enormous impact on the performance of a server. Often we balance fault tolerance against disk reads or disk writes. Regardless of these compromises, the faster you can make the disk subsystem on your computer, the better performance you will see. Examining the disk subsystem does not immediately spring to mind when performance has been questioned, but, as a Windows 2000 professional, you should keep it in the back of your mind. Even though you have plenty of RAM and a fast processor, a disk-intensive application loaded on top of a slow disk subsystem can spell disaster or, at the very least, disappointment.

Network

Networks are fast; networks are efficient—and I have a bridge to sell you in Brooklyn. When examining performance-related issues, you must take everything into account, and you should not assume that the speed of a given network is fast.

Take a 10MB network. That network can theoretically transport 10MB of data per second. A remote office is connected to that network via a T1 connection. A T1 connection has a maximum bandwidth of 1.54MBps. Let's imagine that users on the other end of that T1 line want to connect to a server on the network. Our example is illustrated in Figure 8-3. How much bandwidth do the users have available?

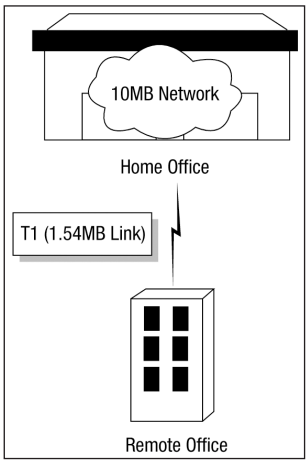

Figure 8-3 Connecting a remote office to the home office

If you think about it for a moment, you imagine that the thinnest pipe, in this case the T1 line, is the limiting factor. So, the answer might well be 1.54MB. But maybe the 10MB network is drowning in data—perhaps it has only 1MB of bandwidth remaining, even if the T1 line is operating at its most efficient. Now, suppose the network card at the server is taking a beating because of a new database application that has been installed, and it's processing data very slowly, effectively cutting the data throughput even more.

As you can see, there is no simple answer to the question, "How much bandwidth is available on the network?" You cannot simply look at a network diagram and get any idea of what is going on. Instead, you must measure bandwidth. This ability is going to be in your arsenal of monitoring tools by the time you reach the end of this chapter.

System Monitor Terminology

When dealing with System Monitor, you basically need to be familiar with three key terms:

- Throughput
- Queue
- Response time

You will see these terms time and again as you work with System Monitor, so it is a good idea to get a clear definition for each of them before we move on.

Throughput

When we talk about **throughput**, we are really talking about how much gets done in a given period of time. As more tasks are thrown at a server, more will get done, and therefore throughput will increase. Then comes the time when the server (or a component on the server) begins to be overwhelmed. At that point, throughput will decrease.

When throughput begins to decrease, either less activity is going on at the server (old requests have been processed and no new requests need attention) or a component is getting behind. This second condition is often known as a **bottleneck**.

Measuring throughput is obviously very important, but do not make any assumptions about the amount of work a component is doing and whether that component is the cause of a bottleneck. The fact is, a component can be humming along 99 percent utilized but still keep up, whereas a hard disk might have one thing to do—say, write a 5GB file—and fall on its knees. Careful monitoring will help keep you on track regarding which components are truly causing problems.

Queue

When a server (or a component in a server) begins to fall behind, things have to wait to be processed. This line of things waiting to be processed is known as a **queue**. When you have a queue, you essentially have a wait state—nothing can proceed until the data is dealt with. This wait can cause performance issues.

Queues can be created by a continuous stream of data hitting a server or server service, when the process simply cannot keep up. They also can be caused by a large amount of data hitting a server at the same time. The data might not be too much to handle if it were broken up into parts, but in bulk it simply overwhelms a system. Zero queues are good; lots of long queues are bad.

Response Time

As you might suspect, **response time** is a measurement of elapsed time from the beginning of a process to its conclusion. Most of the time, you might be looking at small processes; but response time can also be applied to end-to-end processes, such as a request originating in an accounting system and being returned from the server.

Windows 2000 has several different methods of looking at response time. We will examine them later.

Previous versions of Microsoft Windows NT include Performance Monitor, which is a tool very similar to System Monitor. Some might say it is the same program under the guise of a Microsoft Management Console (MMC) snap-in. But there have been some changes, and some of them are significant. What follows applies equally to both versions, though—except where noted.

System Monitor Data Collection

Before we begin really digging into examples of using System Monitor in Windows 2000, we need to work through one more section of definitions in the area of data collection. We'll answer two very good questions: "What can you monitor?" and "How is data collected from a Windows 2000 server?"

First, let's take a look at how System Monitor (and Performance Monitor before it) captures data from a Windows 2000 machine. To answer the first question, data can be collected from something called **system resources**, which consist of memory, processors, disks, and network components. (These components were defined earlier in the chapter.)

Of course, these are very general areas. When you drill down on any one of the resources, you will find all kinds of information. You can obtain data on hundreds of parameters for each of the four general categories.

In addition to the four key areas already specified, it is worth noting that a software vendor can also add parameters that can be monitored. For instance, a database vendor might add a counter that allows an administrator to view the efficiency of queries against a database. This capability equally applies to any service on a Windows 2000 server.

To answer the second question, the parameters that can be monitored are stored in the Registry in a performance key for each service, application, or operating system component. The counters are stored in DLLs that are installed on your system. This chapter will give you a general overview, with some drill-down into the more useful parameters before concentrating in the area of Active Directory.

Each parameter that can be monitored is defined in a performance counter library—a DLL. The Active Directory performance counter library is NTDSPERF.DLL.

You can also use another method to get data supplied to System Monitor, but we will talk about it later in this chapter, in the section "Windows Management Instrumentation."

We're next going to define three key terms that are used to describe how you can configure System Monitor to gather data for you. Once you understand these terms, you will be ready to move on to using the tool. The three terms are:

- Objects
- Counters
- Instances

These terms form a type of hierarchy of objects, which have counters, which in turn have instances. This relationship is shown in Figure 8-4.

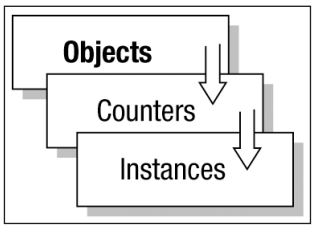

Figure 8-4 The relationship among objects, counters, and instances

Objects

An **object** is a system resource. Some of the most common objects you will use include Network Interface, Paging File, and Processor. There are literally hundreds of objects. You can think of an object as a container for counters and instances.

Any application or service can also appear as an object, so you should not think of objects as being just server or hardware related. Objects can be any measurable item defined within the system.

Counters

A **counter** is used to measure various aspects of an object. The Processor object includes counters for Processor Time, User Time, and Interrupts/Sec, among others. Every object will have at least one counter associated with it, but, as illustrated here, most have more than one.

The trick to successful performance monitoring is figuring out which counters are important in a given situation. We will show you some basic counters in a moment, but this book isn't long enough to discuss every counter in detail.

Instances

An **instance** is a specific counter that is in use. For example, there is an object called Thread. Processes running on a system use threads, and you can monitor each of them by using the Thread object. When you have more than one process running on a system, you can monitor one thread or multiple threads on the system. If you are monitoring Processor Time for two threads, each thread is considered an instance of the counter.

Recommended Counters

Now that you have a good idea of the terms relating to System Monitor, it is time to mention the names of specific counters that can help you troubleshoot issues on a server. Of course, we cannot imagine all the types of issues you might be troubleshooting, or list every available counter. Instead, we will list counters that are most commonly used for each of the four key areas mentioned earlier: memory, processor, disk, and network.

Counters for Memory Troubleshooting

A lack of memory is most evident when a server is beginning to perform poorly (which in simple terms means more slowly). Although this can be an easy issue to identify through other methods (such as Task Manager), a good system administrator should be able to give more detail than simply saying, "All the memory is being used." The counters described in the following sections will help you gather additional data.

Pages/Sec (Memory Object)

Windows 2000 performs a lot of caching, because accessing data that is in physical memory is far faster than fetching it from the hard disk. When a process attempts to gain access to the data that is not held in memory, the result is a **hard page fault**. Hard page faults cause a system to perform more slowly. The Pages/Sec counter helps you track the number of times a hard page fault occurs. The lower this number is, the better. The solution to a high value might be to add more memory, but a high value can also be caused if a slow hard disk causes reads from the disk to be too slow.

Committed Bytes (Memory Object)

If a system is running low of physical memory, it writes out to the hard disk. This process is known as **paging**. A process can actually reserve space in the page file in case it's ever needed; doing so increases performance, should the page file be needed. The Committed Bytes counter records the amount of space that has been reserved.

Pool Nonpaged Bytes (Memory Object)

The physical memory in a Windows 2000 system is assigned specific tasks. One of these areas is the **nonpaged pool**. As suggested by its name, the nonpaged pool contains data that cannot be paged—that is, it must always be in memory while it is being used. Needless to say, because this data cannot be paged, if a lot of data exists in the nonpaged pool you can quickly run out of physical memory. A high reading in the Pool Nonpaged Bytes counter suggests that you need additional physical memory.

Pool Nonpaged Allocs (Memory Object)

The unfortunate term **Allocs** is actually a shortening of **Allocation**. The Pool Nonpaged Allocs counter tells you how many times a request occurs to allocate memory from the nonpaged pool. This counter allows you to determine (in conjunction with the Pool Nonpaged Bytes counter) whether you are making lots of calls to allocate memory from the nonpaged pool. If you are receiving many calls to allocate memory, especially after installing a new application, then the system may have a memory leak.

Available Bytes (Memory Object)

The Available Bytes counter allows you to view quickly how much physical memory is still available. Windows 2000 has a pretty sophisticated way of using physical memory. Say a program loads and needs 1MB of memory. Windows 2000 goes ahead and allocates it. Now, after 10 more processes have started up, the system finds that it doesn't have enough memory to satisfy new requests. First Windows 2000 tries to take memory back from the processes that are already running; if that fails, it begins to page to the hard disk. This process obviously has an impact on the performance of a server. A higher number is better here.

Avg. Disk Queue Length (Physical Disk Object)

A queue is a line of information that is waiting to be processed. Computer processes are busy, and any time they are made to wait is bad for the system. The Avg. Disk Queue Length counter shows the average wait length for both disk reads and writes. The higher this number is, the worse performance is going to be.

Avg. Disk Sec/Transfer (Physical Disk Object)

The Avg. Disk Sec/Transfer counter records the average amount of time for a disk transfer. A **disk transfer** is a transfer of data from memory to disk, or from disk to memory. If you do a little math, you can figure out the percentage of time your system is paging: Simply multiply the value generated by the Memory object counter Pages/Sec (defined above) by this value. For example, a value of 0.2 indicates that paging is taking place 20 percent of the time. This behavior would cause poor performance if it continued for an extended period.

% Usage (Paging File Object)

You may have guessed that the % Usage counter tells you what percentage of the page file is currently in use. What might surprise you is that any value less than 100 percent is acceptable, but higher values are better than lower values. For instance, if you have allocated 100MB to a page file, it is a waste of disk space if you are using only 4 percent of that space for actual paging. Basically, anything up to 99 percent is acceptable—but tuning a system to use 99 percent and never more is extremely difficult.

Pool Paged Bytes (Server Object)

The Pool Paged Bytes counter records the amount of space that has been allocated on the server for paging purposes. The higher this value is, the slower your system will be. However, setting this value too low can cause a system to become virtually inoperable. This counter allows you to see how much of the page file is actually being used. This information is useful when you're optimizing a server. If you install a new application on a server, you should review all the counters in the memory section.

Pool Nonpaged Bytes (Server Object)

The Pool Nonpaged Bytes value will tell you the size of the nonpaged memory pool. We just defined this pool as a memory area where data that must not be paged out to hard disk is stored. If this area of memory is too large, there may not be enough memory for other services and processes. This situation would indicate a need to add more physical memory to the system.

Counters for Processor Troubleshooting

The system processor is, obviously, the heart of your system. A poorly performing processor will bring a system to its knees. The problem is that it is not always clear precisely

what is overwhelming a processor. The most obvious guess would be that too many processes are running on the server (applications, users, and services). But this might not be the case—hardware failures in other components can also cause the processor to max out. The following counters will help you determine how your processor is doing.

Processor Queue Length (System Object)

The Processor Queue Length counter tells you the number of threads waiting to be processed. This counter should never have a sustained value of two or greater. If it does, then you probably need a faster processor in your server.

Interrupts/Sec (Processor Object)

Various processes and pieces of hardware on a server generate interrupts. These interrupts are basically calls to the processor, telling it that the process of a piece of hardware needs attention. It might be that a task is about to begin, or that a task has ended. Such processes should also cause system activity to increase. If the number of interrupts increases, but system activity does not increase, you could have a hardware problem. An example would be a bad network card that is sending a constant stream of interrupts.

% Processor Time (Processor Object)

The % Processor Time counter tells you the percentage of time your processor is in use. A high sustained value means your system needs either a faster processor or an additional processor.

Counters for Disk Troubleshooting

As we mentioned earlier in this chapter, system administrators often jump to assumptions regarding system performance. The first consideration is often the amount of memory. If that is not the problem, then they point to the system processor. You need to be aware of the importance of the disk drives in your system. This component can have an enormous impact (both positive and negative) on how well your system performs. What follows are some counters that can help you track the efficiency of disk subsystem.

Current Disk Queue Length (Physical Disk)

The Current Disk Queue Length counter can be a little tricky because, to make sense of it, you need to know how many spindles exist on your hard disk. Usually there is one; this counter can become complicated when your system has multiple disks, however, especially when they are configured to use RAID 5. (RAID 5 is a set of disks—a minimum of three—that have been configured to act as one.) This counter records the number of system requests that are queued (waiting) to be processed. Its value should never be more than two times the number of spindles in your system. The actual number will vary from server to server, depending on the disk configuration.

% Disk Time (Physical Disk)

As you might have guessed, the % Disk Time counter simply gives you a reading of the percentage of the time your disks are being used. A value of up to 90 percent can be considered good. If the value is higher, you might want to consider a faster disk or a different disk configuration. If your server has a RAID device, this counter can give false readings. You can use % Disk Time in conjunction with Current Disk Queue Length to determine if a disk subsystem is having problems.

Avg. Disk Bytes/Transfer (Physical Disk Object)

The Avg. Disk Bytes/Transfer counter tells you the average amount of data that is transferred from a disk during read and write operations. This value should be greater than 20KB. If it is not, then an application on your system is probably using the disk drive inefficiently. A low value does not necessarily indicate that you need a new hard disk.

Disk Reads/Sec (Physical Disk Object)

Although it is useful to get the big picture view of how well a disk drive is transferring data, averages do not always give you all the information you need. For instance, a hard disk might be overwhelmed with work—but the situation could be caused by a memory problem that is increasing the number of writes to the disk or by an application that is constantly reading from the disk (such as in video streaming). The Disk Reads/Sec counter can identify the rate that disk reads are occurring on a disk.

Disk Writes/Sec (Physical Disk Object)

Disk Writes/Sec is a partner to the previous counter. This counter records the rate at which disk writes are occurring.

Counters for Network Troubleshooting

The network and the network operations on a system can have a direct effect on the perceived speed of a system. If you have a dial-up Internet connection at your office or home, then you know what we're talking about. If you take the fastest system in the world and hook it up to the Internet, but your dial-up connection is slow due to congestion caused by a failure somewhere else, you will begin to think that your system is running slowly. In fact, the memory, processor, and disk subsystem on your system may not be stressed at all—the network connection is causing the slowdown. The following counters will help you determine whether you have a network problem.

Bytes Total/Sec (Server Object)

The Bytes Total/Sec counter is not directly related to memory usage; however, you should periodically check it. This counter records the number of bytes that a server receives from the network and is putting onto the network. If this value is consistently

high, and you are having performance issues, you might want to increase the amount of memory on the server.

Work Item Shortages (Server Object)

As items are received, they need to be processed. The Work Item Shortages counter will increase when incoming requests are being received but no work items are available to process the requests. This situation will cause the network to appear slow.

Bytes Received/Sec (Network Interface Object)

Servers have different roles. It can be useful to know if a server is receiving a lot of data, or whether it is sending data out onto the network. The Bytes Received/Sec counter shows you how much data is being received per second. Compare this to the counter that follows.

Bytes Sent/Sec (Network Interface Object)

The Bytes Sent/Sec counter indicates how much data a server is putting back onto the network. A server filling the role of a read-only database application might have a very high Bytes Sent as opposed to Bytes Received.

Packets Outbound Errors (Network Interface Object)

The Packets Outbound Errors counter tells you how many packets are being discarded at a server because they contain errors. A high number could indicate a problem with the network card. If you have multiple network cards on your machine, then there is an instance for each card; you can check each one.

Counters for Active Directory Troubleshooting

So far, this chapter has concentrated on making sure that you have a good understanding of System Monitor. Without a firm foundation of what it is and how it works, it is impossible for you to use the tool effectively to identify performance and other problems.

We have concentrated on the four main areas with which System Monitor is concerned. These four areas relate to system performance and allow you to determine whether a system has enough memory, whether you have a bad network interface, and so on. It is now time to zero in on how you can use System Monitor to help troubleshoot issues with Active Directory.

When Active Directory is installed on a server, a set of counters is added to System Monitor. These counters work in the same way as other counters. What follows is a list of some significant counters you will use when monitoring the various elements of Active Directory. Once again, it is not possible to list every counter.

DRA Inbound Bytes Total/Sec (NTDS Object)

DRA stands for Directory Replication Agent. The DRA Inbound Bytes Total/Sec counter tells you the sum of the number of bytes received by the DRA for replication. The DRA receives both uncompressed data (for instance, intrasite replication data) and compressed data (such an intersite replication data). However, the counter's sum is for data in uncompressed format—so, in the case of intersite replication traffic, the counter does not necessarily indicate the amount of physical data sent (because the packets are compressed).

DRA Inbound Full Sync. Objects Remaining (NTDS Object)

The DRA Inbound Full Sync. Objects Remaining counter indicates how many objects remain to be replicated when a full synchronization is taking place. This counter can help you determine how long a process is likely to take, and what percentage of objects have already been completed.

DRA Inbound Objects Applied/Sec (NTDS Object)

The DRA Inbound Objects Applied/Sec counter tells you the number of objects that have been received and written to the local Directory Service. These objects are received from replication partners. By looking at this counter, you can quickly find out how many replication updates are taking place due to changes that are being processed by other servers.

DRA Inbound Object Updates Remaining in Packet (NTDS Object)

Replication updates are received in packets. The DRA Inbound Object Updates Remaining in Packet counter tells you how many objects are waiting to be written to the local Directory Service. If this number remains high, then the server is taking a long time to process updates. This situation could indicate problems on the server, and you can use other counters to find out where the problem lies.

DRA Pending Replication Synchronization (NTDS Object)

The DRA Pending Replication Synchronization counter records the number of directory synchronization requests that are currently queued at a server, waiting to be processed. Essentially, this queue becomes a backlog of replication traffic. If this counter begins to climb, it might indicate that the server is slow to process objects. The larger the number, the worse the situation.

We will make configuration changes and use System Monitor in the Real-World Projects that appear at the end of this chapter. System Monitor is a complex tool that provides you with a lot of data. Mastering its use is key to your success as a system administrator of a Windows 2000 network.

System Monitor Logging Options

Before we conclude our discussion of this important topic, we should spend a little time discussing the various options that are available to you when creating logs from System Monitor. System Monitor can provide you with a lot of data, and these configuration options can help you organize it; these options also allow you to zero in quickly on the pieces of data that are most important to you.

Three options are available from within the System Monitor console: Counter Logs, Trace Logs, and Alerts. These options are shown in Figure 8-5. As you can see, the three options are accessible by clicking on Performance Logs and Alerts in the console.

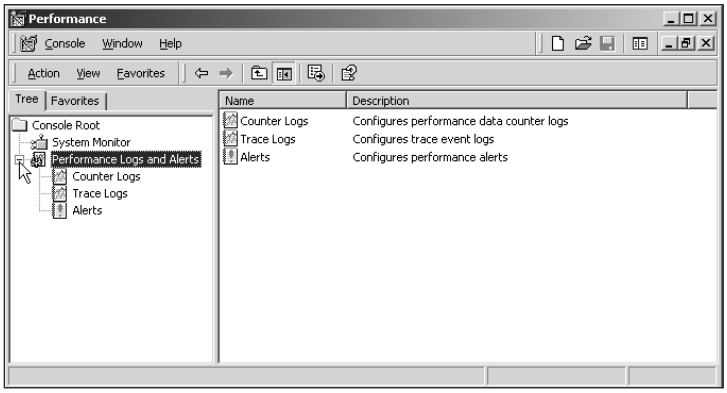

Figure 8-5 The Performance Logs And Alerts option of System Monitor

The three options allow you to set various logging options, such as log names and the format of the log file. We will take a brief look at these options. Before we do, let's define each of them.

Counter Logs

Counter logs are simply ways of creating templates for your performance monitoring tasks. Once you have come up with a group of counters you would like to monitor, you can save them in a template for later use. That way, if you ever want to go back and use the same counters, you do not have to add them one at a time.

The counter logs option dialog box is illustrated in Figure 8-6. The dialog box has various tabs; let's take a closer look at some of the options, so you can see what they do.

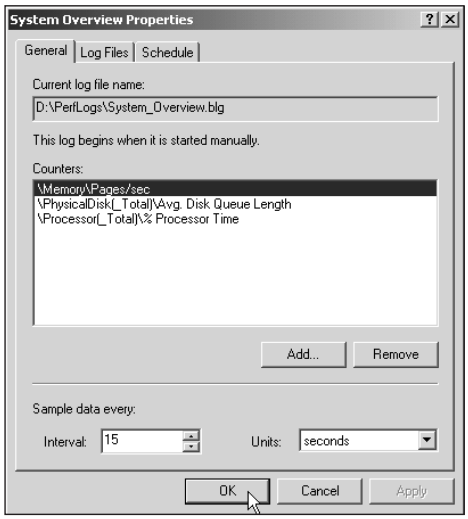

Figure 8-6 The counter logs option dialog box

The General Tab

The General tab allows you to define which counters will be part of this counter log. By clicking on Add, you can add counters to the list.

Other options include the ability to set a sample rate. A **sample rate** is the frequency at which data is sampled. If you want to determine how much data is being transferred to a hard disk, you can set this counter log to sample the transfer rate for a duration of seconds, hours, minutes, or even days. The more frequent the sampling, the more load is put onto the system you are monitoring, because the system now needs to read this data in addition to its existing workload.

The Log Files Tab

Figure 8-7 shows the Log Files tab. The log file allows you to set the properties for the physical file that is used to capture the performance data. These properties include the file location and the filename. Other options allow you to choose the file format; format options include whether the file should be a text format or binary. Text files are easier to use in third-party applications—in fact, you read these log files in Notepad. Binary files require an application to be designed to read them. Of course, System Monitor can read its own binary file format.

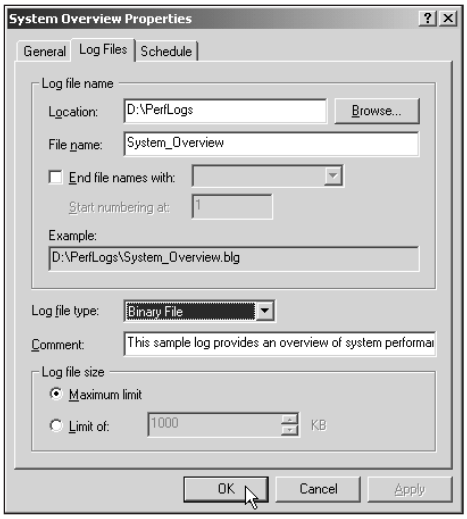

Figure 8-7 The Log Files tab

The final important option on this tab is Log file size. This option allows you to set the maximum size for the log file you are creating. You should consider two things here: Data is constantly being sampled (at the frequency set on the General tab), and the associated log file can grow to be enormous. In fact, unfettered, it would be only a matter of time before the log file filled up the entire hard disk. By default, note that the Maximum Limit option is checked. This option allows the log to grow to the size of available disk space. You should change this setting if you are going to leave logging turned on overnight, or at least make sure you understand the amount of data that your options will generate.

The Schedule Tab

The Schedule tab is shown in Figure 8-8. This tab allows you to configure System Monitor to start gathering data at a specific time. By default, this is a manual process—you configure the options you want to monitor and then tell System Monitor to start. However, this procedure may not always meet your requirements.

Let's say that you want to know the effect on a server of Active Directory replication. The server you are looking at is a bridgehead server between two sites. Because the link between these two sites is busy during the day, the replication is set to take place between 2:00 A.M. and 3:00 A.M. You want to monitor this traffic. Without a scheduling capability, you would have to be at the office at 2:00 A.M. to start the log. With scheduling, however, it is quite easy to set the options you want to monitor, and then have them start automatically.

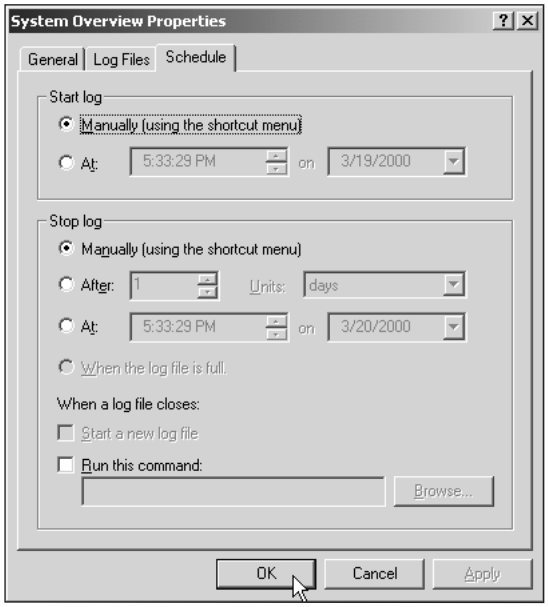

Figure 8-8 The Schedule tab

Another option allows you to decide what will happen when a log file closes. You can either start another log file or execute a program or batch file. The latter choice enables you to have the files copied to another location once they are full, or to perform some other maintenance task.

Trace Logs

Trace logs are a new option that has been added to the monitoring tool since Windows NT 4. Trace logs allow you to configure samples of data to be collected from providers on the system. These providers use Web-Based Enterprise Management (WBEM). WBEM is an industry initiative that defines how data should be collected and also the schema for data collection. We will discuss the architecture of WBEM later in this chapter, in the section "Windows Management Instrumentation."

The data collected from the trace logs is in binary format, and before it can be read it must be parsed. This is another way of saying that you need a special program to read the logs. Unfortunately, Microsoft chose not to include such a tool with Windows 2000. You will have to seek out third-party vendors to use this feature. At the time of this writing, no tools were available.

Alerts

Alerts are a way of making some of the performance data come to life. They do this by allowing you to configure system messages, start a log file, or run an application (perhaps a pag-

ing application to alert you to a significant event). The alert configuration dialog box is shown in Figure 8-9.

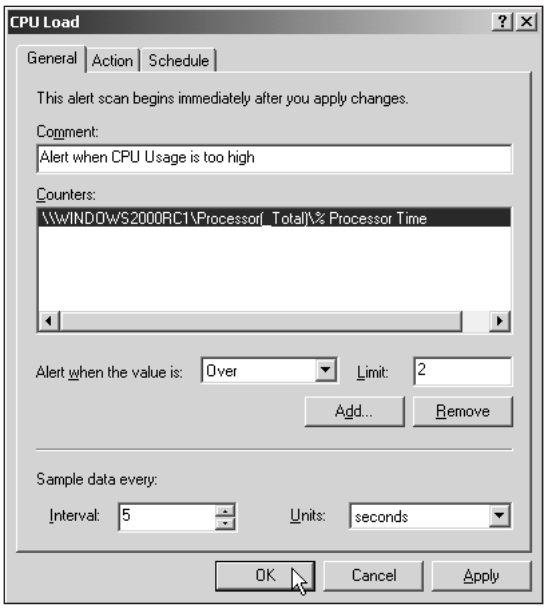

Figure 8-9 The alert configuration dialog box

Alerts can be configured to begin immediately or to occur at specific intervals (such as after work hours). Alerts are essentially triggers; that is, an event will occur if a particular performance monitor counter falls within a certain range. The events that can be triggered include:

- Logging the event to the Application Log
- Sending a network message
- Starting a log file
- Running an external application

System Monitor is a powerful tool that will take some time to master, but it is an important skill that will pay many dividends when you are stuck in front of a server that is not performing correctly. The information in this section will give you a good basis from which to increase your knowledge.

Event Logs

As you have seen in our discussion of System Monitor, you can monitor many different operations on a system. After reading the previous section, you may think that all monitoring takes place in realtime—but this is not the case. Sometimes, it is simply important that events be recorded. You can read the logs to be aware of what has been going on.

Event logs have existed in all versions of Microsoft Windows NT, but you will find that Microsoft has extended their use in Windows 2000. There are now logs for specific functions, which is particularly important for a domain controller in a Windows 2000 environment. We will take a closer look at the logs that are available on a DC, along with some tips on how to read them. You probably will spend a lot of time in the Event Viewer tool, so familiarity is essential to your being an effective administrator.

Event Viewer in Windows 2000 looks a little different from what you might be familiar with from previous versions, as you can see in Figure 8-10. Each available log file is shown in the left pane. The data in the log is shown on the right. Let's take a closer look at the each of these log files, so you understand what data they contain.

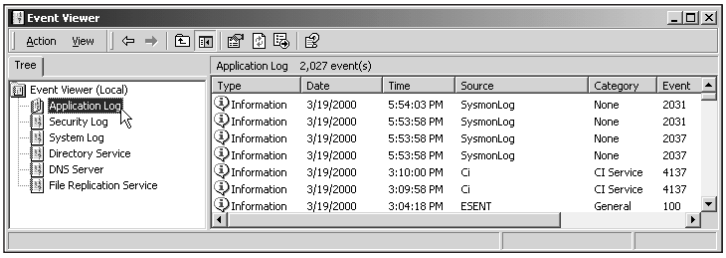

Figure 8-10 Event Viewer

Application Log

The Application Log records events for applications that exist on the system. An example of the use of the Application Log is troubleshooting a failure of an e-mail system or system management application.

The Application Log can quickly become full. In a moment, we will look at how you can control that growth. On standalone systems, it is a safe bet that you will spend much of your time in the Application Log, because most of the day-to-day operations are reported to it. If you are having a problem and want to see if something is wrong, this is the place to start.

Security Log

By default, the Security Log is essentially turned off. The systems are configured by default to be fairly loose—that is, to have a minimal amount of logging occurring.

The Security Log responds to auditing events that are configured via the Computer Management MMC snap-in and Group Policy settings. These options are not turned on by default because they cause a lot of logging to take place. They should be used sparingly, and only when needed. For instance, it is possible to configure a log event to occur every time a user logs on to the network. In an enterprise, this logging will cause a lot of data to be written to the Security Log, a load to be generated at the server, and disk space to be consumed.

You can also audit object access, such as a user's accessing a file or a folder. Doing so might be useful if you think users are trying to access parts of the network from which they have been excluded. Again, you should use these options only when they are needed.

System Log

The System Log is the location to which all messages generated by the system are reported (with the exception of those components that now have their own logs). Messages from applications such as Microsoft Exchange are recorded in the Application Log. The components that report to the System Log include services and components such as Netlogon and the File Replication Service (FRS).

If you are having difficulties starting a service on a system, this log would be a good place to start your search for the cause of the problem. It is worth noting that a system often has dependencies that can cause long strings of errors. For example, if TCP/IP fails on a system, the server and workstation services also fail. This failure, in turn, will cause Netlogon to fail, and so on. The key to troubleshooting such an issue is to locate the operation that caused the dependency to fail and then troubleshoot it from there. The component reporting an error condition might not have a problem at all—the problem may exist elsewhere in the system and be reported elsewhere in the log.

Directory Service

Messages generated by Active Directory have their own log, because the volume of messages would make it difficult to find them all if they were recorded elsewhere. The Directory Service log records events that relate to all aspects of Active Directory operation, including the failure of a replication event or the stopping and starting of the database engine.

This log also records events for the Knowledge Consistency Checker (KCC) and the automatic maintenance processes. (If these terms are unfamiliar to you, you should read Chapter 14, which discusses each of them in detail.)

DNS Server

The DNS Server log records all events that have to do with the Domain Name System (DNS) operations on a DC with the DNS service installed. DNS is an important part of Windows 2000; without it, Active Directory will fail to function.

Because DNS is such an important component, it stands to reason that you will need to monitor it specifically—and that is why DNS now has its own log. You will find entries in this log for such events as being unable to write to Active Directory (because DNS can now be integrated into Active Directory, this occurrence can be fatal).

File Replication Service

The File Replication Service has many different tasks within Windows 2000. One of these tasks is to replicate data between SYSVOLs on DCs. This folder is an essential part of many aspects of Windows 2000, including Group Policy.

Group Policy is not the only time this service is used, however; it is also used for domain Distributed file system (Dfs) and for the replication of source files for software distribution functions of group policy. This log will contain data such as an inability to replicate data from one DC to another. This problem could be due to an error at the server that is instigating the replication (the one reporting the error) or at the destination location.

On standalone and Windows 2000 Professional systems, you will see only the Application, Security, and System logs. On DCs, you will see all six logs, because the role of the server is that much greater. You should understand that the role of system administrators will generally cause them to spend their time in the three most common logs, because they contain data that is more concerned with day-to-day operations (such as application failures).

Event Types

You will see three event types in the various event logs. Each type has its meaning, and you should have different levels of concern regarding them. The three types are:

- Information
- Warning
- Error

Each event type has its own associated icon; these icons make the events easy to see when you are glancing quickly at a full log. These icons are shown in Figure 8-11.

Figure 8-11 The icons for the three event types. You should be most concerned about error messages, because these types of log entries indicate that something has failed.

Table 8-1 gives you a quick definition for what each type of event means, and why the events can be generated. As you will see, many events are purely informational and exist only to let you know that processes are occurring and whether they are successful.

Table 8-1 Event types

Event Type	Description
Information	This type of message records the successful operation of a task or application. For instance, when services successfully start on a system when it is booted, informational messages will appear.
Warning	A warning is an event that may or may not be significant. For instance, when hard disk space begins to get low on a system, the condition is logged as a warning message. It alerts you to the fact that you might have a space problem developing, but for the moment other things are working fine.
Error	Error messages demand immediate attention. They signify a problem that can be anything from the failure of a service to start, to the loss of data. All error messages should be investigated to see how they affect your system. For instance, a single service can cause multiple services to fail. You should seek out errors and eliminate them as soon as possible.

Viewing Remote Event Logs

Event Viewer is a flexible tool. Event logs contain a lot of key information, and you can also view the logs on remote systems. It is simply a matter of connecting to a remote system.

Figure 8-12 shows how you can view the Application Log of a remote machine. As you can see, you simply right-click on the Event Viewer (Local) option and choose Connect To Another Computer. Once you have done this, simply enter the name of the system you want to connect to. As you might imagine, you need sufficient access rights to read another system's logs. For instance, you must have at least administrator access to read the Security Log.

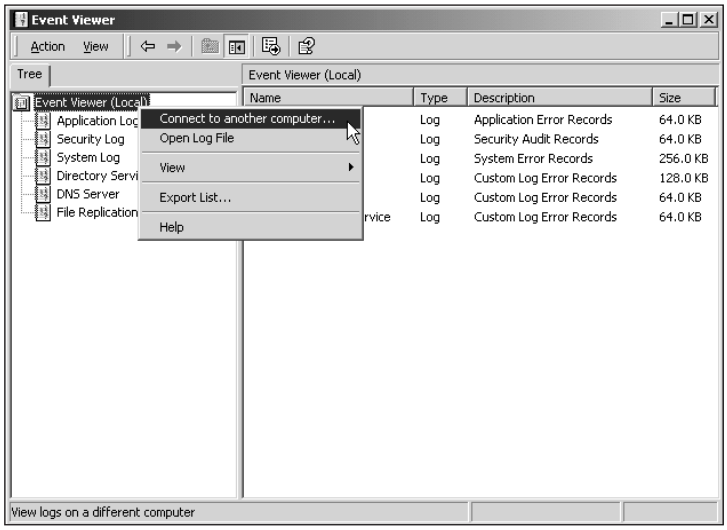

Figure 8-12 Connecting to a remote computer

Managing Event Logs

Event logs are essential to your monitoring techniques; however, although the data is undoubtedly important, you should make sure that you are only keeping data that will be useful. Several options are available when it comes to configuring the log files; they apply to each log equally. You can:

- Provide an upper size limit for the log files
- Set the number of days that data should be kept
- Save the events for archival purposes

Although it might seem like a good idea to collect data for as long as you can, the truth is that you are not going to have time to go through endless logs looking for significant events. It is far better to put some time aside each day to view the event logs and see what has occurred. If you do this regularly, then there will be no need to keep archives going back weeks or months.

The exception would be if you are trying to discover trends on a new server that has been installed on your network. It can be a good idea to monitor a new server a little more closely, to make sure that it is completely functional. Logs for these types of servers can be archived for later analysis.

The Log Properties dialog box is shown in Figure 8-13. As you can see, several options are available. These include the maximum size of the log file (by default, 512K) and actions to take if the log file reaches its maximum size.

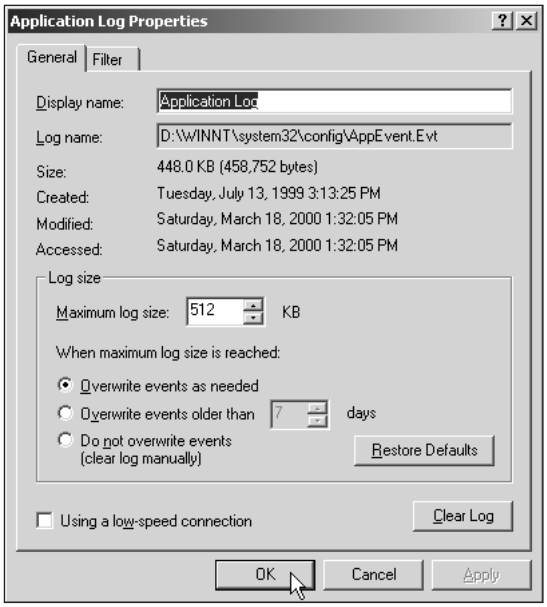

Figure 8-13 The Log Properties dialog box

These options are described in the following:

- *Overwrite Events As Needed*—Causes events to be deleted from the log once it has reach its maximum size. This happens regardless of the number of days an event has been in the log. If a lot of logging is occurring in a system, it is possible that events from the same day could be overwritten. In systems that are used less, it could be days or weeks before events are overwritten.
- *Overwrite Events Older Than*—By default, set to seven days. With this option set, events older than the specified number of days will be overwritten, even if maximum log size is not reached. This option can be useful when you never want to review log data that is older than a given length of time.
- *Do Not Overwrite Events (Clear Log Manually)*—Causes events to be never deleted from the log file. This option can cause problems if you forget you have set it and the log file fills up. (Windows 2000 will stop logging events if this occurs, with the exception of security events.) This is also the most secure of all the options, because no event is ever lost (overwritten or deleted).

The Event Viewer application is an important tool that you can use to monitor the actions and functions occurring at a particular system. Because the Security Log can contain information regarding security breaches, you should pay close attention to the data it contains. Viewing these log files should become part of your everyday activities as a system administrator.

Windows Management Instrumentation

A key new feature has been added to System Monitor (it was not available in previous versions of Window NT). This feature is the integration between System Monitor and the Windows Management Instrumentation (WMI). WMI is a key industry initiative, and it is worth spending a few paragraphs defining it and explaining why it is so significant.

WMI first made an appearance in a significant way with the release of Microsoft Systems Management Server 2.0 (SMS). Earlier versions of SMS were fairly good at getting hardware inventory data from clients on a network. The amount of data was arbitrary and limiting, however. Also, some of the most important data (such as serial numbers of computers) could not be collected.

Along came Intel and many other hardware vendors, who decided that there should be a way to pull data from a computer. Not only that, but if the method were industry defined, then the information from (say) a Compaq machine ought to be the same as the information from a Dell.

Sometimes, however, things don't work out the way everyone wants. This group of vendors did come up with something called the Desktop Management Interface (DMI), which is basically an Application Programming Interface (API) that allows generic (or at least well-known) function calls to be made to gather data. DMI worked well, but it fell down a little when it came to the agents that had to be installed to make it work. An agent was needed because some piece of code had to run on each machine to make those API calls; and, over time, getting agents from all manufacturers for all machines was time consuming. In addition, the data they returned was not always the same.

Over time, networks have become more complex. Many more things need to be monitored, and the agent piece of the equation had to be simplified. The industry got together again, and came up with a standard on which monitoring and data collection can be based: Web-Based Management Instrumentation (WBEM). Microsoft agreed to implement WBEM on its operating systems, but it also changed the name. Microsoft's implementation is known as Windows Management Instrumentation (WMI).

Because DMI was dogged by agent issues, that problem required a solution. Microsoft came to the rescue by shipping in every copy of Windows 2000 what is essentially the middleware between the applications that display data and the underlying data providers. In fact, Microsoft went a step further and added it to Windows 9.*x* and Windows NT 4, as well. (Service Pack 4 for Windows NT 4 installs the necessary components.)

Now, all that is needed is for vendors to write the piece that collects the data. In WMI terms, this is a combination of providers and Management Object Files (also known as MOF files). Some of these have already been installed in your copy of Windows 2000.

You will find the WMI files in the <*Systemroot*>\system32\Wbem folder. The MOF files can be found in <*Systemroot*>\system32\Wbem\Mof. You should not delete these files from your system—it will become unstable if you do. You can read more about DMI at **http://www.dmtf.org/**. For information on WMI, go to **http://www.microsoft.com/management**.

WMI extends to many places that DMI never did. For instance, Intel has demonstrated systems that can detect whether a screw had been removed from the back of a case, or if a BIOS chip begins to heat too rapidly. It remains to be seen what will happen to WMI, but we expect it to play a significant role in future generations of reporting and monitoring tools. You read it here first!

Just for clarity, Figure 8-14 presents the architecture for WMI. In the figure, you can see the role of an application, the middleware, and the providers that supply the data. System Monitor is an application like any other, utilizing the data providers.

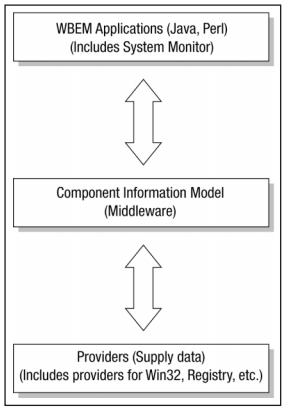

Figure 8-14 WMI architecture

To allow System Monitor to use WMI counters to display data, simply choose Start|Run and type "perfmon /WMI". Press Enter, and System Monitor will start.

Tools for Monitoring Active Directory

Along with the standard set of tools that exists in every installation of Windows 2000, some command-line and Graphical User Interface (GUI) tools are optional. These tools are for the advanced Active Directory administrator and are not to be taken lightly. It is worth mentioning them here, however, so you have a well-rounded view of the available tools.

We will take a brief look at some of the additional monitoring and informational applications available on the Windows 2000 CD. To use some of these tools, you must first install them—

they are not installed by default. To install these tools, you need the Windows 2000 CD. When you insert this CD, you are presented with the Microsoft Windows 2000 CD dialog box (if autorun is not enabled on your system, you will not see this screen; you will have to navigate to the necessary folder manually). Click on Browse This CD to open the folder for the CD contents. Navigate to the SUPPORT\TOOLS folder. Once there, execute the SETUP.EXE program to install the tools we are about to discuss.

When the installation has completed, you will have a new program group called Windows 2000 Support Tools. You can start most of these tools from this group. When there is an alternative method of starting them, we will discuss it in the definition.

What follows are brief descriptions of some of the most important tools. We won't spend too much time on this topic, because these tools are rarely used and are intended for advanced users.

LDAP Diagnostic Tool (LDP.EXE)

The Lightweight Directory Access Protocol (LDAP) diagnostic tool allows you to run LDAP queries against the Active Directory. LDAP is a protocol, and queries within a Windows 2000 tree are made via this protocol. If you are having problems performing queries against the directory on your network, then you can use this tool to confirm that LDAP is functioning. You can perform add, modify, search, and compares on data within the directory.

The graphic tools within Windows 2000 do not necessarily let you see every object that is stored within Active Directory. This includes metadata that is included with some objects. Tools like LDP allow you to expose this information. However, unless you are intimately familiar with the underlying data, these objects might not have much meaning.

Unlike many of the other tools, there is no inline help for this tool. You must also have a firm grasp of LDAP syntax before the tool will be much use. Unfortunately, a detailed discussion of LDAP is beyond the scope of this book. A typical LDP display showing Active Directory data is shown in Figure 8-15. To start this tool, follow the installation directions given at the beginning of this section. Then choose Start|Run, type "LDP", and click on OK.

Active Directory Replication Monitor (REPLMON)

The Active Directory Replication Monitor (REPLMON) allows you to view the status of replication between partners. You can also view some performance information. This application offers a lot of information, such as the directory partitions stored on a server and which direct replication partners are relevant to each. It will display both direct partners and also partners through transitive trusts. Icons within REPLMON quickly show you whether a partner is operational.

Along with replication data, REPLMON can display the roles that a server is playing in the enterprise. For instance, if a server is also the PDC Emulator, this status would be displayed.

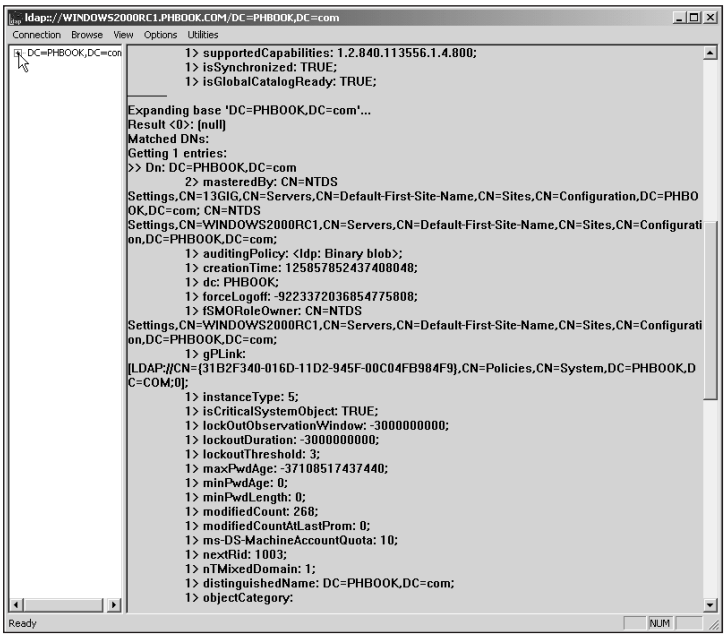

Figure 8-15 The LDP tool

For direct replication partners, you can see the Update Sequence Numbers (USNs) of the servers and the number of failed replication attempts that have occurred. You can also view the Globally Unique Identifier (GUID) for the partner and the protocol the partner is using to replicate data.

Finally, you can trigger replication manually within this tool. Doing so might be necessary if you know a server is out of date, or if you want to monitor the effect of replication traffic on a particular segment of your network or at a DC.

REPLMON is a powerful tool, and despite all the good points mentioned here, you can do a host of other things with it. We suggest (once you have mastered everything else in this book) that you take a closer look at REPLMON. When you want to have a better understanding of replication in your enterprise, there isn't a better tool.

REPADMIN

Just because it does not have a fancy name, don't be fooled into thinking that the REPADMIN command-line tool is not powerful. (In fact, Windows 2000 has introduced a whole host of command-line tools for many functions. If you are coming from a Unix environment, you might very well be more comfortable working outside of the standard GUI tools provided by Microsoft.) This utility can display the names of replication partners. This information is useful when you want to know how a server fits into the overall replication topology.

The one drawback of this command is that it does not see the connection objects created in Active Object. Instead, it shows only the connections made by the KCC. (If you do not remember what these terms mean, go back to Chapter 6 for a review.)

The list of command-line switches available within REPADMIN is shown in Figure 8-16. As you can see, many options are available for you to use. For example, open a command prompt on a DC, type "REPADMIN /showrep", and press Enter. Doing so will display the replication partners for the server according to the KCC.

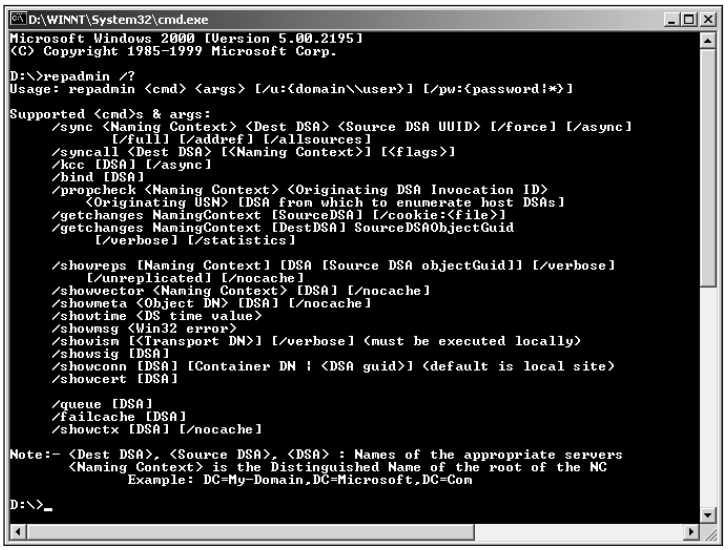

Figure 8-16 REPADMIN command-line switches

These tools, and many others, are useful when troubleshooting and monitoring Active Directory and Windows 2000. It would take an entire book to discuss each of them in detail. Once you have learned everything in this book, you will be ready to tackle the other tools. Here, we have presented a good overview of the tools, both for real-world use and for the Microsoft 70-217 exam.

Chapter Summary

- In this chapter, we looked at methods you can use to monitor the performance of domain controllers on your network. Some of the tools and strategies can also be used on Windows 2000 Professional systems, but they will not work on Windows 9.*x* machines.
- We first examined what constitutes a system that is performing poorly. It is fairly common to underutilize systems and to aim for unrealistic goals (such as 0 percent CPU usage). We discussed all the major steps administrators take to improve a system,

such as adding a new hard disk, memory, or processor. We also warned against simply throwing additional resources at problems without first performing some kind of analysis.

- We introduced the term **baseline** to describe what normal operation should be for a system. A baseline is a set of statistics that describes the performance level for a system on a normal working day. You use a baseline as a standard, and then compare subsequent statistics against it to identify problems or to compare the performance of a system after a change has been made.
- The first built-in tool we looked at is System Monitor. System Monitor is an MMC snap-in that allows you to access counters for various components (also known as **objects**) available on a system. System Monitor is a new and improved version of the Performance Monitor that shipped with Microsoft Windows NT.
- You should be concerned about four main areas when performing baseline analysis or examining a system for variations from the baseline: memory, processor, disk, and network. These areas work together to determine the overall performance of a system. System Monitor gives you the ability to collect statistics for each of these key areas. In System Monitor, these areas are known as *objects*. Many other objects are available, but these are the basic four.
- Memory includes statistics for the RAM inside your machine (including usage and amount free) and counters for the swap file. The swap file is an area of hard disk space that acts like RAM, should the system require additional memory space.
- The Processor object includes counters that allow you to see how busy the system processor is. You can view this information as a percentage of capacity (processor is working at 90 percent of capacity, for instance) or in more sophisticated ways by viewing the number of requests waiting to be serviced by the processor. If your system has more than one processor, you can view each of them individually.
- The Disk object allows you to view a large range of information, including the amount of space free on a disk, how quickly data is being written to the disk, and how quickly the disk is able to perform reads. When performing system monitoring of disks, you should take into consideration the fact that your servers are likely to be using a fault-tolerant system (such as RAID 5)—such a system can skew the results.
- The Network object concerns itself with the performance of the communication components on a system, including the amount of data being put onto the wire by the system and being read off the wire. If a system appears to be performing poorly, it might be because data cannot be written to or from the network quickly enough.
- Performance of a system is usually discussed using the following three terms: throughput, queue, and response time. Throughput measures the amount of work that takes place in a given period of time. Queue defines the line of waiting requests that need to be processed. Response Time measures how quickly a process is completed (a measurement of start to finish).

- Each of the four objects discussed (Memory, Processor, Disk, and Network) has statistics you can use to measure performance. These statistics are called objects, counters, and instances.
- An object is a representation of a system resource. Each object has associated counters and instances. The object level is the catch-all where you can view every statistic available in a particular area. It is worth noting that a complete analysis of a system will include analysis of data from many different object types.
- A counter is a specific area of statistics for a given object. For instance, the Processor object has counters for User Time and Interrupts/sec. A specific counter can generate statistics for every occurrence of a resource in a system, via instances. In order to measure a counter, you must measure an instance.
- An instance is a specific occurrence of a counter. A counter can have multiple instances. For instance, the Processor object has a counter that generates statistics for the percentage of time a processor is being used. This counter is the % Processor Time. If your system has more than one processor, there will be multiple instances of this counter for you to monitor.
- System Monitor in Windows 2000 includes hundreds of counters—far too many for us to list in this book. As well as supporting the built-in counters, it is also possible (and probable) that third-party vendors will add their own objects and counters to System Monitor. We listed some of the most commonly used counters in each of the four areas we have identified. These counters should be used before a problem occurs, to define the baseline for a system.
- These counters work together to form a description of a system and how it performs. We also discussed counters that are specific to Active Directory. This is an example of a vendor adding counters to System Monitor to help you view the performance of a newly installed service (in this case, Directory Services).
- System Monitor data is very important, and you should run it periodically so you can compare results over time. Rather than setting up the options individually each time, you can configure counter logs. Counter logs allow you to choose counters once and then reuse them. The results can be saved to a file on disk for later analysis, and you can choose from several different file formats. Once you have configured the options for your counter logs, you can schedule when (and how often) the counters should run.
- System Monitor works by reading data from the system Registry. Each counter is defined in a Dynamic Link Library (DLL) on the hard disk. System Monitor also allows you to use WMI (Microsoft's implementation of WBEM). WBEM is an industry-defined method of collecting data. It helps define which data can be collected, and how it should be presented.
- Another important tool is Event Viewer. Event Viewer allows you to view the event logs that are recorded on every Windows 2000 system. The number of logs will vary, depending on the services and functions installed on a system. The three standard

logs that appear on every system are the Application Log, Security Log, and System Log. In addition, on Windows 2000 DCs, you will find a Directory Service log, DNS Server log, and File Replication Service.

- Each of these logs displays data on various events. There are three event types: information, warning, and error. These should elicit varying levels of concern. Information messages are probably not of concern; error messages mean something has failed, and should therefore be examined carefully.
- Event logs can become very large (and the more logs on a system, the more space is used). You can set options for event logs that include the maximum size of the logs, the number of days data is stored in the log, and how events should be saved.
- Finally, we took a brief look at some tools that are available specifically for monitoring Active Directory. These tools are often rather obtuse and complex, and are not installed by default. They include the LDAP Diagnostic Tool, the Active Directory Replication Monitor, and REPADMIN. You can use these tools to view the details of Active Directory replication or to determine if you have problems with queries to the directory.